

UNITED STATES PATENT APPLICATION

FOR

**METHOD AND SYSTEM FOR SYNCHRONIZING MULTIMEDIA I/O WITH  
CPU CLOCK**

Inventors:

Rainer W. Lienhart

Igor V. Kozintsev

Dmitry N. Budnikov

Igor V. Chikalov

Sergey A. Egorychev

Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

12400 Wilshire Boulevard

Los Angeles, CA 90025-1026

(408) 720-8300

"Express Mail" mailing label number: EV336409362246US

Date of Deposit: 12-31-03

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450

Carla Vignola

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

# **Method and System for Synchronizing Multimedia I/O with CPU Clock**

## **NOTICE OF RELATED APPLICATIONS**

**[0001]** This application is related to Application Serial No. \_\_\_\_\_, entitled “ Method and System for Synchronizing Platform Clocks In A Distributed Wireless Platform”, filed on December 31, 2003, Attorney Docket No. 42390.P18599, and Application Serial No. \_\_\_\_\_, entitled “Method and System for Synchronized Distributed Audio Input on Gernal Purpose Computer Platforms”, filed on \_\_\_\_\_, Attorney Docket No. 42390.P18597, which applications are assigned to the assignee of the present application.

## **COPYRIGHT NOTICE**

**[0002]** Contained herein is material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction of the patent disclosure by any person as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all rights to the copyright whatsoever.

## **FIELD OF THE INVENTION**

**[0003]** The present invention generally relates to the field of distributed multimedia synchronization. More particularly, an embodiment of the present invention relates to synchronizing multimedia I/O with the CPU clock.

**[0004]** One approach to provide additional computing power has been to utilize distributed computer environments. This approach enables several computers to collaboratively perform computational tasks within a reduced amount of time. Generally, the divide and conquer approach provided by such parallel computing approaches enables utilization of available personal computers, rather than purchasing of a high performance, server-based computer system for performing the computationally intensive tasks.

**[0005]** Distributed computing has generally, however, been applied to performing purely computational tasks and not to synchronized capture and/or processing of signals, especially audio/video signals (and data streams). Signal processing of audio/video signals (and data streams) are generally very sensitive to even very small differences in sampling rates (e.g., clock skew), jitter, and delays. Therefore, precise synchronization is very critical for high quality input/output processing, as well as for real-time performance and in general, robustness and reliability issues. But, precise capture and synchronized inputs are not guaranteed on current platforms.

**[0006]** For example, on the same personal computer (PC) platform, problems can arise when several input/output (I/O) devices are used to capture audio and visual information from video camera(s) and microphone(s). Due to the fact that the different I/O devices will be triggered by separate oscillators, resulting audio samples and video frames will not be aligned on an absolute time line (thus inducing some relative offsets). Moreover, due to differences in the oscillators' frequencies, audio and/or visual data will drift away across multiple channels/streams over time.

Instabilities in the oscillators' frequencies will also not be perfectly correlated between each other.

**[0007]** Similarly, in the case of multiple PC platforms audio and visual I/O devices will not be synchronized in time scale inducing some relative offsets and data samples to drift relative to each other. The extent of the relative offset, drift, and jitter on the existing platforms depends on many hardware and software parameters and can be very significant, sometimes causing total degradation of the processed signals (from the non-synchronized input streams). Such drifts, delays, and jitters can cause significant performance degradation for instance for array signal processing algorithms.

**[0008]** For example, in an acoustic beam former with 10 centimeter (cm) spacing between microphones, an error of only 0.01 percent in time can cause error of 20 degrees in the beam direction. Due to this fact, current implementations of audio array process algorithms may rely on dedicated circuitry for the synchronization between multiple I/O channels. Unfortunately, implementing such an approach with existing PC platforms would require a major overhaul of the current hardware utilized by the PC platforms. Therefore, there remains a need to overcome one or more of the limitations in the above-described existing art.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0009]** The invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar or identical elements, and in which:

**[0010]** **Fig. 1** illustrates an exemplary block diagram of a computer system 100 in which one embodiment of the present invention may be implemented;

**[0011]** **Fig. 2** further illustrates the I/O devices 200 of the computer system 100 as depicted in **Fig. 1**;

**[0012]** **Fig. 3a** illustrates a General Purpose Computing platform with a main CPU clock and separate clocks on each peripheral device;

**[0013]** **Fig. 3b** illustrates a diagram of a system for processing multimedia streams;

**[0014]** **Figs. 4a-b** illustrate a system module to synchronize multimedia streams, in accordance with one embodiment; and

**[0015]** **Fig. 5** illustrates a flow diagram describing the processes to synchronize a multimedia stream, in accordance with one embodiment.

## **DETAILED DESCRIPTION**

**[0016]** In the following detailed description of the present invention numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

**[0017]** Reference in the specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

**[0018]** Also, the use of the term general purpose computer (GPC) herein is intended to denote laptops, PDAs, tablet PCs, mobile phones, and similar devices that can be a part of a distributed audio/visual system.

**[0019]** **Fig. 1** illustrates an exemplary block diagram of a computer system 100 in which one embodiment of the present invention may be implemented. The computer system 100 includes a central processing unit (CPU) 102 coupled to a bus 105. In one embodiment, the CPU 102 is a processor in the Pentium® family of processors including the Pentium® II processor family, Pentium® III processors, Pentium® IV processors available from Intel Corporation of Santa Clara, California. Alternatively, other CPUs may be used, such as Intel’s XScale processor, Intel’s Pentium M Processors, ARM processors available from ARM Ltd. of Cambridge, the

United Kingdom, or OMAP processor (an enhanced ARM-based processor) available from Texas Instruments, Inc., of Dallas, Texas.

**[0020]** A chipset 107 is also coupled to the bus 105. The chipset 107 includes a memory control hub (MCH) 110. The MCH 110 may include a memory controller 112 that is coupled to a main system memory 115. Main system memory 115 stores data and sequences of instructions that are executed by the CPU 102 or any other device included in the system 100. In one embodiment, main system memory 115 includes dynamic random access memory (DRAM); however, main system memory 115 may be implemented using other memory types. Additional devices may also be coupled to the bus 105, such as multiple CPUs and/or multiple system memories.

**[0021]** The MCH 110 may also include a graphics interface 113 coupled to a graphics accelerator 130. In one embodiment, graphics interface 113 is coupled to graphics accelerator 130 via an accelerated graphics port (AGP) that operates according to an AGP Specification Revision 2.0 interface developed by Intel Corporation of Santa Clara, California. In an embodiment of the present invention, a flat panel display may be coupled to the graphics interface 113 through, for example, a signal converter that translates a digital representation of an image stored in a storage device such as video memory or system memory into display signals that are interpreted and displayed by the flat-panel screen. It is envisioned that the display signals produced by the display device may pass through various control devices before being interpreted by and subsequently displayed on the flat-panel display monitor. The display device may be a liquid crystal display (LCD), a flat panel display, a plasma screen, a thin film transistor (TFT) display, and the like.

**[0022]** In addition, the hub interface couples the MCH 110 to an input/output control hub (ICH) 140 via a hub interface. The ICH 140 provides an interface to input/output (I/O) devices within the computer system 100. In one embodiment of the present invention, the ICH 140 may be coupled to a Peripheral Component Interconnect (PCI) bus adhering to a Specification Revision 2.1 bus developed by the PCI Special Interest Group of Portland, Oregon. Thus, the ICH 140 includes a bus bridge 146 that provides an interface to a bus 142. In one embodiment of the present invention, the bus 142 is a PCI bus. Moreover, the bus bridge 146 provides a data path between the CPU 102 and peripheral devices.

**[0023]** The bus 142 includes I/O devices 200 (which are further discussed with reference to **Fig. 2**) and a disk drive 155. However, one of ordinary skill in the art will appreciate that other devices may be coupled to the PCI bus 142. In addition, one of ordinary skill in the art will recognize that the CPU 102 and MCH 110 may be combined to form a single chip. Furthermore, graphics accelerator 130 may be included within MCH 110 in other embodiments.

**[0024]** In addition, other peripherals may also be coupled to the ICH 140 in various embodiments of the present invention. For example, such peripherals may include integrated drive electronics (IDE) or small computer system interface (SCSI) hard drive(s), universal serial bus (USB) port(s), a keyboard, a mouse, parallel port(s), serial port(s), floppy disk drive(s), digital output support (e.g., digital video interface (DVI)), and the like. Moreover, the computer system 100 is envisioned to receive electrical power from one or more of the following sources for its operation: a power source (such as a battery, fuel cell, and the like), alternating current (AC) outlet (e.g.,



through a transformer and/or adaptor), automotive power supplies, airplane power supplies, and the like.

**[0025]**        **Fig. 2** further illustrates I/O devices 200 of the computer system 100 as depicted in **Fig. 1**. As illustrated, the computer system 100 may include a display device 212 such as a monitor. The display device 212 may include an intermediate device such as a frame buffer. The computer system 100 also includes an input device 210 such as a keyboard and a cursor control 208 such as a mouse, trackball, or track pad. The display device 212, the input device 210, and the cursor control 208 are coupled to bus 142. The computer system 100 may include a network connector 206 so that computer system 100 may be connected as part as a local area network (LAN) or a wide area network (WAN) such as, for example, the Internet.

**[0026]**        Additionally, the computer system 100 can also be coupled to a device for sound recording and playback 230 such as an audio digitization device coupled to a microphone for recording voice input for speech recognition or for recording sound in general. The I/O devices 200 of computer system 100 may also include a video digitizing device 220 that can be used to capture video images alone or in conjunction with sound recording device 230 to capture audio information associated with the video images. Furthermore, the input devices 200 may also include a hard copy device 204 (such as a printer) and a CD-ROM device 202. The input devices 200 (202-212) are also coupled to bus 142.

**[0027]**        Accordingly, the computer system 100 as depicted in **Fig. 1** may be utilized to capture multimedia data including, for example, audio and/or video data from a selected scene, environment, or the like. Currently, many individuals utilize

personal computers (PCs) such as depicted in **Fig. 1** in order to capture live audio/video data (multimedia scene data) through, for example, a camera coupled to a port of computer system 100 (not shown) such as, for example, a USB port or a firewire port (IEEE 1394). This data is then provided as a streaming media format (Multimedia Stream Data) including, but not limited to, Microsoft® advanced steaming format (ASF) files, motion picture experts group (MPEG) standards such as MPEG-1/2/4, and audio layer-3 (MP3) files, Real Audio G2 files, QDesign2 files, or the like.

**[0028]** In one embodiment of the present invention, an audio capture device such as a microphone may be utilized by the computer system 100 to capture audio information associated with the captured multimedia scene data. Accordingly, as individuals attempt to utilize their personal computers in order to capture, for example, live audio/video data, it is generally recognized that audio/video data is most effectively captured utilizing one or more data capture devices.

**[0029]** With reference to **Figs. 1 and 2**, the I/O devices (except AGP display adapters) are generally connected to the ICH (I/O hub) via dedicated or shared buses. The PCI bus can be a way to connect various audio, video, and networking devices to the ICH. These devices typically have their own crystal oscillators and clocks that are not synchronized to each other, and to the CPU clock. This means, for example, that if audio and video samples are captured using separate I/O cards, they can go out of sync as time passes by.

**[0030]** Unfortunately, the time it takes for a block of data to travel between I/O device, main memory, and CPU is variable and depends on many factors like the

CPU load, cache state, activity of other I/O devices that share the bus, and the operating system behavior. Therefore, applications that process data have no way to know precisely the time the data enters or leaves the I/O devices. The propagation delay may range from nanoseconds to milliseconds depending on the conditions mentioned above.

**[0031]** In existing applications, multiple video and audio streams are usually captured using a single I/O device such as a multi-channel analog to digital (A/D) or audio/video (A/V) capture cards. Special methods are needed to use multiple I/O devices synchronously even on a single PC platform.

**[0032]** The situation becomes more complex when synchronization of I/O devices on separate platforms is desired. There, in addition to I/O-CPU latencies, network connection introduces additional delays, that are variable due to best-effort (and therefore variable transmission delay) type of Media Access Protocols used in existing wired and wireless Ethernet.

#### Overview of the Synchronization Variations

**[0033]** **Figure 3a** illustrates a typical GPC platform with the main CPU clock 332 and separate clocks 334, 336, 338, 340 on each peripheral device. In one embodiment, to provide I/O stream synchronization a linear transition model is generated for each peripheral device, which relates the stream offset (that is the sample number in the audio stream) to the value of the CPU clock counter (RDTSC). As described herein, I/O stream and multimedia stream are referenced interchangeably.

[0034] In one embodiment, let  $t$  be a value of the CPU clock and  $\tau_i$  the sample number that was recorded/played at time  $t$  with the  $i$ -th device. The model then has the following form:  $t(\tau_i) = a_i(t) \tau_i + b_i(t)$ , where  $a_i(t)$  and  $b_i(t)$  is the timing model parameter for the  $i$ -th device. In one embodiment, the dependency of the model parameter on time approximates instabilities in the clock frequency due to temperature variations and other factors. In one embodiment, the model can have the inverse form:  $\tau_i(t) = a_i(\tau_i) t + b_i(\tau_i)$ ,  $t$  is a value of the system clock and  $\tau_i$  is a sample number of the multimedia stream at time  $t$  with a  $i$ -th device, and  $a_i(\tau_i)$  and  $b_i(\tau_i)$  is the timing model parameter for the  $i$ -th device. In one embodiment, the synchronization of the I/O stream is divided into two parts: learn transition model coefficients  $a_i(t)$  and  $b_i(t)$ , and shift and resample streams according to transition model.

[0035] A brief description is provided describing the synchronization operations and timing relationships on a GPC during audio capture, as illustrated in **Figure 3b**, in accordance with one embodiment. Audio output would follow a similar routine.

[0036] Incoming audio packets are received and processed by a hardware device 306 (e.g., network card), and eventually is put into a Direct Memory Access (DMA) buffer 308. The time for the hardware component is modeled in **Figure 3b** by the delay  $d_{hw}$ , which is approximately constant for similar hardware modules.

[0037] A DMA controller transfers the data to a memory block allocated by the system and signals the event to the CPU by an Interrupt ReQuest (IRQ). The stage issuing the IRQ introduces variable delay due to memory bus arbitration between different agents (i.e., CPU, graphics adapter, other DMA's).

[0038] The interrupt controller (APIC) 310 queues the interrupt and schedules a time slot for handling. Because APIC is handling requests from multiple devices, this stage introduces variable delay. Both previous stages are modeled by  $d_{\text{isr}}$  in

**Figure 3b.** The Interrupt Service Routine (ISR) of the device driver 312 is called, and the driver 312 sends notification to the Operating System (OS) 314. The OS delivers a notification and data to the user application(s) 316.

[0039] As described above, the data packet traverses multiple hardware and software stages in order to travel from network adapter to the CPU and back. The delay introduced by the various stages is highly variable making the problem of providing a global clock to the GPCs a very complicated one.

#### Synchronizing Multimedia device and CPU Clock

[0040] A description of synchronizing a multimedia device and a CPU clock, in accordance with one embodiment, is provided. In one embodiment, the ISR of the multimedia driver timestamps samples in the OS buffer using the CPU clock to form a set of observation pairs  $(\tilde{t}^j, \tau_i^j)$ . In one embodiment,  $j$  represents the index of the transferred multimedia buffer.  $\tilde{t}^j$  represents the value of the CPU clock at the time of the  $j$ -th transfer, and  $\tau_i^j$  represents the

sample number of the last/first sample in the buffer that was transferred at time  $t^j$  for the  $i$ -th device for input/output, in accordance with one embodiment.

[0041] In accordance with the description accompanying **Figure 3b**,  $\tilde{t}^j$  can be obtained from  $\tilde{t}^j = t^j + d_{hw} + d_{isr}$ , which may further be modeled as  $\tilde{t}^j = t^j + d + n$ . In one embodiment,  $d$  models all constant delay components and  $n$  represents the stochastic component. Given the set of observations  $(\tilde{t}^j, \tau_i^j)$ , in one embodiment, an estimate is generated for the timing model parameters  $a_i$  and  $b_i$  for all peripheral devices.

[0042] In one embodiment, the values of  $a_i$  and  $b_i$  are generated using a least trimmed squares (LTS) regression. In one embodiment, LTS is equivalent to performing least squares fit, trimming the observations that correspond to the largest residuals (i.e., defined as the distance of the observed value to the linear fit), and then computing a least squares regression model for the remaining observations.

#### Streams Adjustment and Resampling

[0043] Using the parameter estimation techniques described above, in one embodiment, I/O streams may be synchronized (i.e. start simultaneously and have the same sample rate). The synchronization of the I/O streams may be performed by pre- or post-processing of the stream. **Figure 4a** illustrates a system module to synchronize I/O streams to be played back, while **Figure 4b** illustrates a system module to synchronize I/O streams that are being captured, in accordance with one

embodiment. The modules, in one embodiment, consists of I/O hardware 410a-b; model estimator 412a-b; a re-sampler 414a-b that changes sampling frequency of stream; a gate 418a-b that is to pass-through or reject samples; and an application 428a-b that works with the synchronized stream.

**[0044]**        **Figure 5** presents a flow diagram describing the processes to synchronize an input I/O stream, in accordance with one embodiment. The processes of the flow diagram of **Figure 5** are described with reference to the module of **Figure 4b**. In process 502, the application 418b issues a start command and specifies the start time in CPU clock units. In process 504, the I/O device 410b starts capturing the multimedia stream ahead of time, but samples do not pass to the application 418b as the gate 416b is closed. In process 506, the estimator 412b collects observations (i.e., time stamp pairs) until the number of observations collected becomes sufficient for estimating model parameters.

**[0045]**        In process 508, given CPU time, capture start time, and model parameters, an offset, in the multimedia stream corresponding to the first sample of the synchronized multimedia stream requested by application 418b, as well as the re-sampling coefficients, are calculated. In process 510, the gate 416b is opened when the offset in the I/O stream reaches offset value calculated in process 508. In process 512, the resampler 414b changes sampling frequency of the I/O stream according to the resampling coefficient calculated in process 508.

**[0046]** A related set of processes would be used to synchronize I/O streams that are to be played back, with appropriate variations to the processes for the operation of playing the I/O stream. In addition, in the case of one or more applications requesting multiple multimedia streams, in parallel, the streams may all be synchronized with the CPU clock, in accordance with the description above.

**[0047]** Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. For example, although much of the description herein references the multimedia stream as audio, the techniques described herein would also apply to video streams. Therefore, references to details of various embodiments are not intended to limit the scope of the claims which in themselves recite only those features regarded as essential to the invention.